

### REMARKS

Reconsideration and allowance of the above identified patent application are hereby requested. Claims 9, 11-13, 15, 17, 18, 27, 29-31, 33, 35, 36 and 47-53 are now in the application with claims 9, 27, and 47 being independent. Claims 9, 27, and 47 have been amended. Claims 37-46 have been canceled without prejudice or disclaimer in response to their withdrawal from consideration and subject to Applicants' right to pursue the subject matter of the canceled claims in a divisional application. The Office's rejections are respectfully traversed.

#### **Rejection Under 35 U.S.C. §103(a)**

Claims 9, 11, 12, 15, 17, 18, 27, 29, 30, 33, 35, 36, 47-49, and 51-53 stand rejected under 35 U.S.C. §103(a) as allegedly being unpatentable over U.S. Patent No. 6,961,750 to Burd et al. in view of U.S. Patent No. 6,507,867 to Holland et al. Further, claims 13, 31, and 50 stand rejected under 35 U.S.C. §103(a) as allegedly being unpatentable over Burd et al. in view of Holland et al. and further in view of U.S. Patent No. 6,470,349 to Heninger et al. These contentions are respectfully traversed.

#### Claim 9:

Claim 9 recites (emphasis added) "...locating, at the client, a start identifier and an end identifier in the information received from the server, wherein the end identifier corresponds to the start identifier; identifying, at the client, a first object associated with at least one of the start identifier and the end identifier, wherein the first object comprises server side code stored at the

client; and storing, at the client, an item of information appearing between the start identifier and the end identifier in association with the first object, wherein the item of information comprises a result generated by the server in accordance with the first object.”

The Office (Action of December 12, 2007 at pages 2-5) asserts that the proposed combination of Burd et al. and Holland et al. discloses the claimed subject matter. For example, the Office (*Id.* at pages 3-4) asserts that (emphasis added)...

In summary, Burd teaches a server sending a client HTML code, wherein the HTML code can contain server-side code (col. 1, lines 64-67 through col. 2, lines 28, col. 5, lines 14-18). An example of the server-side code that Burd refers to could relate to the display of the dynamic changing of a stock price or the display of the constantly changing traffic information. This server-side code can be formatted into the HTML, that the client computer identifies in order to view the webpage with the content on a client computer. In HTML files, there are HTML start and end tags that appear before and after HTML code (For example, Fig. 4, items 2, 3, 10, 11, 12, 19, 20, 21, etc.). Server-side code (as well as other code in the HTML file) is located, identified, and associated by the client. The “result” is the HTML data that is generated by the server and that gets transmitted to the client (Fig. 2, 210, 214) for displaying (Fig. 2, 216) of the web page (which includes the displaying of the changing stock prices, traffic updates, etc.) on the client using a web browser (col. 1, lines 64-67 through col. 2, lines 28).

Although, the Office (*Id.* at page 4) concedes that Burd et al. do not disclose storing at the client, an item of information appearing between the start identifier and the end identifier in association with the first object, the Office asserts that Holland et al. and Burd et al. together disclose such storing. Nonetheless, the proposed combination of Burd et al. and Holland et al. fail to disclose the claimed subject matter.

Claim 9 has been amended to further clarify that server-side code is stored at the client. In contrast, Burd et al. do not disclose that server-side code is stored at the client. For example, Burd et al. (Abstract) disclose (emphasis added) “A server-side control object processes and generates a client-side user interface element for display on a web page.” Further, Burd et al. (Col. 4, lines 60-65) disclose that (emphasis added)...

In an embodiment of the present invention, server-side control objects logically correspond to client-side user interface elements and generate at a server the authoring language code to be used by a client-side browser to display and process a web page.

Thus, Burd et al. teach that the server-side control object generates code for display at the client. Burd et al., however, do not disclose that server-side code is provided to or stored at the client.

The Office (Action of December 12, 2007 at page 3) also asserts that (emphasis added) “Burd teaches a server sending a client HTML code, wherein the HTML code can contain server-side code (col. 1, lines 64-67 through col. 2, lines 28, col. 5, lines 14-18).” However, the Office's assertion is not supported by Burd et al. For example, Burd et al. (Col. 1, line 64 through Col. 2, line 2) disclose (emphasis added)...

In some circumstances, a web page may need to display dynamic content in a browser, such as a changing stock price or traffic information. In such situations, a server-side application program is typically developed to obtain the dynamic data and format it into HTML, that is sent to the browser for display in a web page as the web page is updated.

Thus, Burd et al. teach that an application at the server sends an updated HTML page to a client for display. However, Burd et al. do not teach that the HTML page includes server-side code.

Rather, the web page is updated by the server sending new HTML code to the client. Burd et al. (Col. 2, lines 31-38) also disclose (emphasis added)...

The serve[r] side application program can process the HTTP request and generate the appropriate HTML code for web page with a newly computed calendar to reflect the user's action for transmission to the client in an HTTP response. Thereafter, the resulting document is transmitted to a client system in an HTTP response, where it is displayed in the browser as a web page that shows the updated calendars.

Again, Burd et al. teach that an application at the server generates HTML code for display at the client. However, Burd et al. do not teach that the HTML page includes server-side code. Moreover, Col. 5, lines 14-18 of Burd et al., to which the Office cites, also disclose that a client receives and displays HTML code from a server, but fails to indicate that the HTML code includes server-side code.

Additionally, the Office (Action of December 12, 2007 at pages 3-4) cites to operations 210, 214, and 216 of FIG. 2. Operation 210 is performed by the server, while operations 214 and 216 are performed by the client. With respect to those operations, Burd et al. (Col. 7, lines 52-58) disclose (emphasis added)...

Operation 210 transmits the HTML code to the client in an HTTP response. In operation 214, the client receives the HTML code associated with a new web page to be displayed. In operation 216, the client system incorporates (e.g., displays) the user interface elements of the new page in accordance with the HTML code received from the HTTP response.

Thus, Burd et al. teach that the server transmits HTML code representing a web page to the client. However, Burd et al. do not disclose that the HTML code includes server-side code. To

the contrary, Burd et al. (Col. 7, lines 63-67) disclose (emphasis added) “In an embodiment of the present invention, server-side code objects in the hierarchy are created in response to an HTTP request...and destroyed subsequent to the rendering of authoring language data (e.g., HTML data).” Thus, Burd et al. teach that server-side code objects are not sent to the client along with HTML code, but rather are destroyed after the HTML code is rendered. Accordingly, Burd et al. do not disclose that server-side code stored at the client.

Further, the Office does not demonstrate how Burd et al. disclose that server-side code is associated with at least one of the start identifier and the end identifier. The Office (Action of December 12, 2007 at page 9) asserts that (emphasis added) “In HTML files, there are HTML start and end tags that appear before and after HTML code...and HTML code could include objects such as the server-side objects mentioned in Burd.” Assuming *arguendo* that HTML code includes both HTML start/end tags and server-side objects, which is not conceded, Burd et al. nonetheless fail to disclose an association between server-side code and a start or end tag. Accordingly, Burd et al. do not disclose, teach, or suggest identifying, at the client, a first object associated with at least one of the start identifier and the end identifier, wherein the first object comprises server side code stored at the client, as recited in claim 9.

The proposed combination of Burd et al. and Holland et al. also fails to disclose storing, at the client, a result generated by the server in association with the first object (server side code). The Office (Action of December 12, 2007 at page 9) asserts that Burd et al. teach (emphasis added)...

associating, at the client, an item of information appearing between the start identifier and the end identifier with the first object, wherein the item of

information comprises a result generated by the server (server generates the HTML code that contains the object before sending to client to be displayed on web browser) in accordance with the first object (col. 1, lines 64-67 through col. 2, lines 1-39, Fig. 2, items 214, 216, col. 7, lines 52-67).

Thus, the Office does not assert that an item of information comprising a result generated by the server and appearing between the start identifier and the end identifier is associated, at the client, with the first object. Rather, the Office incorrectly asserts that the HTML code appearing between the start identifier and the end identifier contains the first object.

Burd et al. (Col. 1, line 64 through Col. 2, line 39) disclose that dynamic data, such as a changing stock price or traffic information, can be formatted into HTML by a server-side application and sent to a browser for display. Further, Burd et al. (Col. 7, lines 52-67 and Fig. 2) disclose that the client receives HTML code associated with a new web page and displays the elements of the new web page in accordance with the HTML code. However, Burd et al. do not disclose or suggest that any portion of the HTML code is associated, at the client, with an object that comprises server side code stored at the client. To the contrary, as discussed above, Burd et al. do not disclose that server side code is stored at the client.

The Office (Action of December 12, 2007 at page 4) further asserts that Holland et al. teach that web pages can be stored on a client computer. Nonetheless, Holland et al., taken alone or in combination with Burd et al., do not disclose the claimed subject matter. Rather, the cited portions of Holland et al. (Col. 4, lines 33-43) disclose (emphasis added)...

Another technique is the use of servlets, which are executable code objects that can be dynamically invoked by the Web server to process a user request. Servlets

typically perform some specialized function, such as creating page content based on dynamic factors.

Holland et al. (Col. 9, lines 10-14) further disclose that (emphasis added) “The server side of the implementation may execute on a computer functioning as a Web server, where that Web server provides services in response to requests from a client using a Web browser connected to the Internet.”

As with Burd et al., Holland et al. do not disclose that a result generated by the server in accordance with a first object (server side code stored at the client) is stored at the client in association with the first object. Rather, Holland et al. teach storing at a client one or more web pages, but do not disclose performing the storing such that there is an association with server-side code. Accordingly, neither Burd et al. nor Holland et al. disclose storing, at the client, an item of information appearing between the start identifier and the end identifier in association with the first object, wherein the item of information comprises a result generated by the server in accordance with the first object, as is recited in claim 9.

For at least these reasons, claim 9 is allowable over the proposed combination of Burd et al. and Holland et al. Claims 11-13, 15, 17, and 18 depend from claim 9. Therefore, dependent claims 11-13, 15, 17, and 18 are allowable for at least the reasons discussed with respect to claim 9.

Further, claims 27 and 47 include subject matter similar to that discussed with respect to claim 9. Therefore, independent claims 27 and 47 are allowable over the proposed combination of Burd et al. and Holland et al. for at least the reasons discussed with respect to claim 9. Claims

29-31, 33, 35, and 36 depend from claim 27, and are thus allowable based at least on claim 27.

Claims 48-53 depend from claim 47 and are thus allowable based at least on claim 47.

Claim 13:

Claim 13 recites (emphasis added) “The method of claim 12 wherein the start identifier comprises an even number and the end identifier comprises an odd number, wherein the value of the end identifier is greater than the value of the start identifier.” Additionally, claim 9, from which claim 13 indirectly depends, recites that the end identifier corresponds to the start identifier.

The Office (Action of December 12, 2007 at page 7) concedes that Burd et al. in view of Holland et al. fails to teach the claimed subject matter. Nonetheless, the Office (*Id.*) asserts that (emphasis added)...

Heninger teaches sequentially ordered tokens for each element of a file, such as an HTML file, wherein said elements comprise HTML tags. Therefore, it would have been obvious to one of ordinary skill in the art to modify Burd in view of Holland's HTML tag identifiers such that they would be ordered sequentially (equivalent to the start identifier comprising an even number and the end identifier comprising an odd number, wherein the value of the end identifier is greater than the value of the start identifier).

Nonetheless, Heninger et al. also fail to disclose the claimed subject matter. Also, it is presumed that “Holland's” was intended to be “Heninger's” with respect to the HTML tag identifiers.

Heninger et al. do not teach that tokens are “HTML tag identifiers” as the Office asserts. Rather, Heninger et al. (Col. 6, lines 21-26) disclose that (emphasis added)...



When a script file is initially loaded 70, the elements of the file are parsed by a series of parsers which recognize and process individual types of commands and their related expressions. The HTML parser 72 distinguishes between HTML tags and inventive script commands in the file and creates tokens for contiguous HTML blocks.

Thus, Heninger et al. teach that script commands are distinguished from HTML tags. However, Heninger et al. do not teach that a token is used to identify an HTML tag. Moreover, Heninger et al. teach that a token is created for a contiguous HTML block. A token that is associated with a contiguous block of HTML code cannot be both a start identifier that comprises an even number and an end identifier that comprises an odd number, as recited in claim 13.

Further, Heninger et al. (Col. 6, lines 5-8) disclose that the tokens are sequentially ordered. Contrary to the Office's assertion (Action of December 12, 2007 at page 7), however, sequentially ordering tokens is not equivalent to numbering start and end identifiers. For example, Heninger et al. do not teach a second token that corresponds to a first token. Rather, Heninger et al. teach that a token represents a contiguous HTML block. Further, Heninger et al. do not disclose that a token includes an internal reference, such as a number. As such, the tokens disclosed by Heninger et al. are referenced solely by their position in a sequence. Therefore, Heninger et al. do not disclose or suggest that a start identifier comprises an even number and an end identifier comprises an odd number, wherein the value of the end identifier is greater than the value of the start identifier, as recited in claim 13.

Additionally, claim 13 depends from claim 9 which recites (emphasis added) "...locating, at the client, a start identifier and an end identifier in the information received from the server,"

wherein the end identifier corresponds to the start identifier....” Heninger et al. do not teach that tokens are sent to a client. To the contrary, Heninger et al. teach that tokens are “stripped” from a file before the file is provided to a client. For example, Heninger et al. (Col. 2, line 61 – Col. 3, line 7) disclose (emphasis added)...

When a browser requests a document enhanced by the present invention, the interpreter processes the document by scanning for the inventive script commands. When a script command is encountered, the interpreter executes the script command, typically by retrieving or processing database information, returns the result to the web page and strips the executable script command from the resulting “pure-format” document. Once all executable script commands have been executed and stripped, the resulting document is in a “pure-format” form. In the case of a typical web page, the “pure-format” form would be a pure HTML document containing no extraneous tags or commands. This “pure-format” document is, then, sent by the server to the client computer whose browser initiated the document request.

Heninger et al. (Col. 6, lines 15-18) further disclose (emphasis added) “Execution of the token sequence 64 creates a new ‘pure format’ or ‘pure-HTML’ document that is sent to the client browser that requested the document 32.” Thus, Heninger et al. teach that the “sequentially ordered tokens”, which the Office incorrectly equates with a start identifier comprising an even number and an end identifier comprising an odd number, are stripped from the document and thus never provided to a client. Instead, only a resulting “pure-format” document is sent to the client. Accordingly, no motivation for modifying Burd et al. in view of Heninger et al. can be found in the references, and Heninger et al. actually teach away from the proposed combination. The proposed combination would not provide for “easy recognition of HTML tags”, as the

Office suggests, because Heninger et al. teaches sending only a “pure-format” or “pure-HTML” document to a client browser.

For at least these reasons, claim 13 is patentable over the proposed combination of Burd et al., Holland et al., and Heninger et al. Further, claims 31 and 50 include subject matter similar to that discussed with respect to claim 13. Therefore, claims 31 and 50 are allowable for at least the reasons discussed with respect to claim 13.

Claim 15:

Claim 15 recites (emphasis added) “The method of claim 9 additionally comprising: locating a special attribute identifier in the information received from the server; identifying a second object associated with the special attribute identifier; and storing information corresponding to the special attribute identifier in association with the second object.”

The Office (Action of December 12, 2007 at pages 5-6) asserts that Burd et al. teach (emphasis added)...

locating a special attribute identifier in the information received from the server (col. 12, lines 17-31, col. 9, lines 50-53, col. 12, lines 6-34, col. 13, lines 1-20); identifying a second object associated with the special attribute identifier (col. 12, lines 17-31, col. 9, lines 50-53, col. 12, lines 6-34, col. 13, lines 1-20); and associating information corresponding to the special attribute identifier with the second object (col. 12, lines 17-31, col. 9, lines 50-53, col. 12, lines 6-34, col. 13, lines 1-20).

Burd et al. fail to disclose the claimed subject matter.

Burd et al. (Col. 12, line 6 through Col. 13, line 20) disclose “server-side control objects” and “custom server-side control objects” that can be “associated with a unique ‘id’ attribute to enable programmatic referencing of the corresponding control object.” However, Burd et al. (Col. 9, lines 40–45) teach that the file including the server-side control objects and custom server-side control objects is processed at a server to render HTML code for transmission to the client. For example, Burd et al. (Col. 9, lines 40–65) disclose that the server-side control objects are invoked by a page compiler that “creates and invokes the appropriate server-side control objects” and combines rendered HTML code for transmission to the client. However, Burd et al. do not disclose that any of the server-side control objects are transmitted to the client. Moreover, the Office (Action of December 12, 2007 at page 4) has conceded that (emphasis added) “Burd is silent in storing said associated item of information at the client.”

Therefore, Burd et al. fail to disclose locating a special attribute identifier in the information received from the server; identifying a second object associated with the special attribute identifier; and storing information corresponding to the special attribute identifier in association with the second object, as recited in claim 15. Further, Holland et al. fail to cure the deficiencies of Burd et al. For at least these reasons, the subject matter of claim 15 is patentable over the proposed combination of Burd et al. and Holland et al. Further, claims 33 and 51 include subject matter similar to that discussed with respect to claim 15. Therefore, claims 33 and 51 are allowable for at least the reasons discussed with respect to claim 15.

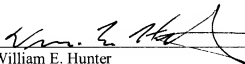
### Concluding Comments

The foregoing comments made with respect to the positions taken by the Examiner are not to be construed as acquiescence with other positions of the Examiner that have not been explicitly contested. Accordingly, the above arguments for patentability of a claim should not be construed as implying that there are not other valid reasons for patentability of that claim or other claims.

In view of the above remarks, claims 9, 11-13, 15, 17, 18, 27, 29-31, 33, 35, 36 and 47-53 should be in condition for allowance, and a formal notice of allowance is respectfully requested. Please apply any other charges or credits to deposit account 06-1050.

Respectfully submitted,

Date: March 12, 2008

  
\_\_\_\_\_  
William E. Hunter  
Reg. No. 47,671

Fish & Richardson P.C.  
PTO Customer No. 21876  
Telephone: (858) 678-5070  
Facsimile: (858) 678-5099  
10816966.doc